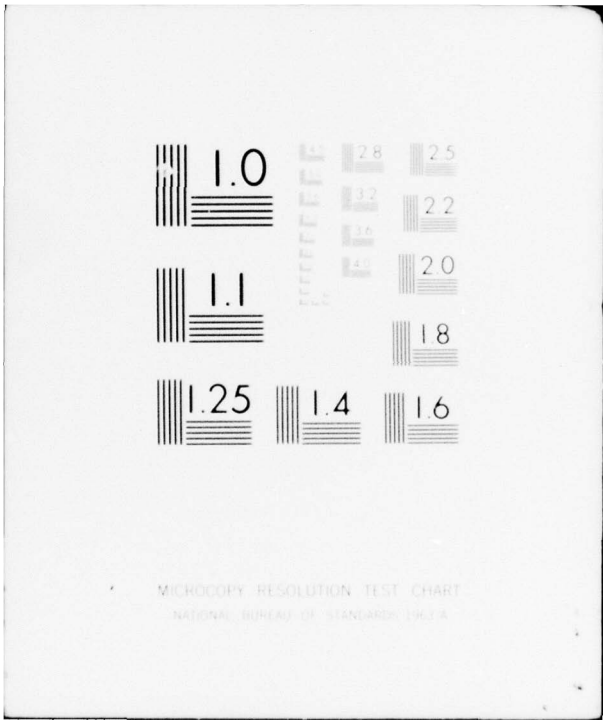
AD-A044 394 NAVAL SURFACE WEAPONS CENTER WHITE OAK LAB SILVER SP--ETC F/G 9/2
COMPUTER AIDED WIRE WRAPPING OF INTEGRATED CIRCUIT BOARDS.(U)
MAY 76 D L BARTUSEK

UNCLASSIFIED NSWC/WOL/TR-76-37 NL

1 OF 1
AD A044394

END
DATE FILMED
10-77
DDC


1.0
1.1
1.25
1.4
1.6
2.8
2.5
3.2
2.2
3.6
2.0
4.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

NSWC/WOL/TR 76-37

12
B.S.

NSWC TECHNICAL REPORT

WHITE OAK LABORATORY

COMPUTER AIDED WIRE WRAPPING OF INTEGRATED CIRCUIT BOARDS

BY
Donald L. Bartusek

MAY 1976

NAVAL SURFACE WEAPONS CENTER
WHITE OAK LABORATORY
SILVER SPRING, MARYLAND 20910

- Approved for public release; distribution unlimited

NAVAL SURFACE WEAPONS CENTER
WHITE OAK, SILVER SPRING, MARYLAND 20910

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: (14) NSWC/WOL/TR-76-37
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): (6) Computer Aided Wire Wrapping of Integrated Circuit Boards.
5. TYPE OF REPORT & PERIOD COVERED: (9) final rept.,
6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): (10) Donald L. Bartusek
8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Surface Weapons Center, White Oak, Silver Spring, MD 20910
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: (11)
12. REPORT DATE: May 1976
13. NUMBER OF PAGES: 24
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): (12) 26p.
15. SECURITY CLASS. (of this report): unclassified
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report): Approved for public release; distribution unlimited
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
18. SUPPLEMENTARY NOTES
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Wire wrapping
Daisey chains
Chain length minimization
Integrated circuit boards
Computer programs

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report is written as a user's manual for the computer algorithm that automatically produces a deck of cards to submit to an automatic wire wrapping facility. A simplified user language is used for data input which is written in dip coordinates taken directly from the logic drawings.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

391596

1B

May 1976

NSWC/WOL/TR 76-37

Computer Aided Wire Wrapping of Integrated Circuit Boards

This report was written as a user's manual to aid the designer of hard wired logic boards. The work was done in support of wire wrapping requirements of the Mk 541 test set AN/UYK-20 upgrade project.

The author wishes to thank Mr. J. M. Pierce and Mr. A. C. Jackson for their helpful discussions while the algorithms were being developed.

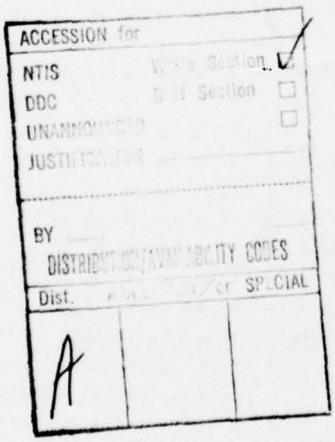


Table of Contents

Table of Contents (continued)

Chapter 1

INTRODUCTION

Two similar computer programs were written to aid in the wire wrapping of integrated circuit boards. One routine, called GNRLWRAP, handles general purpose boards up to two segments wide. The other, called PLUGWRAP, handles plug or socket type boards up to six segments wide. To use the routines, the user must punch an input data check which is written in free field notation. Either routine will handle the Augat type board. The GNRLWRAP routine, in addition, handles the Amp type.

The main purpose of the routines is to machine punch an input drive deck which can then be submitted to an automatic wire wrap facility. The user may, however, specify a hand wire list instead of the normal drive deck.

It is the author's understanding that several companies offer automatic wire wrapping service. Raytheon, for instance, does in their Waltham, Massachusetts plant. Their mailing address and phone number is as follows:

Raytheon Company
Wire Wrapping Facility
20 Seyon Street
Waltham, MA 02150
(617) 899-8400 X2660/3444

The drive decks generated by these routines conform to the Raytheon format. Consequently, they might require modification if submitted to another manufacturer.

The report is concerned with the use of the routines and will not bore the reader with page after page of Fortran listings. Chapter 2 deals with the PLUGWRAP routine and Chapter 3 discusses the GNRLWRAP routine. The chapters are self-sufficient with some repetition.

Chapter 2

PLUGWRAP

This routine specifically addresses the Augat 8136PG5-30 to 8136PG5-180 series boards. The 8136PG5-30 is a single width board holding up to 30, 16-pin dips. The boards are available in multiples of 30 dips (up to 6-wide with 180 dips).

To assemble a deck application for the Naval Surface Weapons Center (NAVSURFWPNCEN) CDC 3200, the user needs the following:

1. Job card [$JOB,1,XXXX,ND] beginning in column one where XXXX may be the users identification.
2. Binary deck called PLUGWRAP
3. Run card [$RUN] beginning in column one
4. Data deck (described below)
5. End of file card [$\begin{smallmatrix}77\\88\end{smallmatrix}$EOF]
6. Endscope card [$ENDSCOPE]

Data Deck

The data deck consists of simple code word instructions along with node to node interconnection data. But first, a few general comments.

1. All input is free field with one or more spaces used as the only delimiter. All eighty columns of the card are read.
2. Since spaces act as a delimiter and terminate a field, none can occur within a field except in the case where a name field has been started.
3. A space in column one signals a continuation of the previous card. There is no limit to the number of continuation cards. The "end of card" also terminates a field and, therefore, a single node reference may not be split between two cards, i.e., breaks can occur only between nodes.

4. The order of the cards is unimportant with the exception that continuation cards must follow, in any order, behind the lead card they continue.

5. All notation concerning node connections refer to pin numbers of the dip rather than pin numbers of the socket.

6. The square brackets, [] , will be used in this report to indicate what is punched on the card, i.e., everything between the brackets is punched on the card. In some cases however, the brackets may be eliminated where the meaning is clear (such as the sample input data deck at the end of this chapter).

Data Deck Options

The following options or input cards are available to the user.

[NAMEDECK XXXXXXXXX]

This card is used by the user who wishes to name a Raytheon drive deck. The nine characters denoted by the X's will be punched in columns 72 to 80 of the Raytheon drive deck. The first nonspace character on or after column 9 begins the 9-character X field.

[NAME XXXXXXXXXXXX?YYY ...,]

This card allows the user to name his circuit design. The first nonspace character on or after column 5 begins the field which extends to the end of the card but is limited to 72 characters maximum. The character string will appear on all output. The Raytheon drive deck, if requested, has room for only the first 12 or 13 characters, however.

[NO PUNCH]

This card is self explanatory and is used by the user who wishes to check the output results before the actual punching of the Raytheon drive deck. The single space between the "O" and "P" is required.

[RAYTHEON DECK]

This card requests a raytheon drive deck output format rather than the hand wire listing (default mode). The routine recognizes only the first eight characters and then ignores or skips over the next five. Therefore, just the word RAYTHEON would suffice.

[DIPXX Dip1 Dip2 Dip3 ...,]

This card is used to list those dips that are other than 14-pin (default mode). The double character XX-field will contain the number of pins of all dips listed on the card. Spaces may occur between the P and XX-field, if desired. Numbers larger than 16 call an error routine. Following the XX-field, the appropriate dips would be listed (separated from each other by one or more spaces).

[GND Node1 Node2 Node3 ...,]

This card may never be used by the user of PLUGWRAP, but is listed for completeness. It was intended for users of the GNRLWRAP routine discussed in Chapter 3. However, it does work and ties all listed nodes individually to the closest available GND pin.

[VCC Node1 Node2 Node3 ...,]

This card may never be used by the user of PLUGWRAP, but is listed for completeness. It was intended for users of the GNRLWRAP routine discussed in Chapter 3. However, it does work and ties all listed nodes individually to the closest available VCC pin.

[SCRATCH Dip1 Dip2 Dip3 ...,]

This card will find use for those occasions where a discrete component carrier is used. The routine assumes that all dip positions listed on the card will not want pins 16 and 8 of the socket tied to VCC or GND. Therefore, these pins are normally removed from the available list of ground and voltage connections that can be made. The user is not required, but is then free to physically scratch out the solder connections that tie these pins to the voltage and ground planes of the printed circuit card. There is one exception, however, and it occurs when a smaller size dip is inserted in the socket. For 16-pin dips, the result is as described above. However, if the dip has less than 16 pins, pin 8 of the socket is allowed as an external ground connection and therefore, the user is not free to scratch it out.

One additional feature associated with the scratch card is that all dips not listed (default mode) will have their half-pin numbers tied to ground. Thus, all 14-pin dips that are not listed on a scratch card will have pin 7 tied to ground. There is one exception, however, to this rule and that occurs when the half-pin number is included in a daisey chain that is ungrounded.

Consequently, the user can think in terms of pin numbers of the dip or discrete carrier. If he does not want the highest pin number to be tied to VCC or the half-pin number to be tied to GND, he should then list that dip on a scratch card. He may scratch out the solder connection to pin 16 of the socket in all cases. The solder connection to pin 8, however, may be scratched only if the dip is a 16-pin dip.

As an afterthought, the scratch routine was modified so that the user could restrict the scratch operation to either the voltage or ground connection on the dip. A "G" or "V" punched immediately after the dip entry causes the above discussion to apply only to the appropriate connection on the dip. A space after the dip entry signals the routine to treat both connections as originally intended.

[DAISEY Node1 Node2 Node3 ...,]

This card supplies the basic interconnection data to the routine, and since it will probably be used the most, it has been made the default mode of code words. In other words, no code word is also a daisey card as follows.

[Node 1 Node2 Node3 ...,]

Here the daisey card is assumed. All nodes listed on the card (and its continuation cards) will be chained together in the most optimum order (minimum total wire length). If this optimum order is not desired, another form of the Daisey card is available (see Hold-card described below).

[HOLD Node1 Node2 Node3 ...,]

On this card the listed nodes will be held "as is" and will not be reordered. This will make the card useful in generating documentation that agrees with handmade corrections. It might even be desirable to generate a new input drive deck so that reordered boards will be carbon copies of the corrected prototype.

A slight complication exists if a ground or voltage connection is required. On any daisey card (which includes the hold card as an alternate form) ground or voltage connections are made by punching [G] or [V] on the card as a valid node input. In fact, any character string, without numbers, that has G or V as a first character, will be accepted. Thus, the user may prefer [GND] or [VCC]. The problem arises from the fact that the ground or voltage connection may be made from either end of the chain. To specify which end, the user punches enclosing parentheses around the first or last node of the chain, and the ground or voltage

connection will be restricted to that end. The user is cautioned that on standard daisey cards (nonhold form) the order of the nodes may be permuted by the routine. Therefore, if the node enclosed by parentheses does not end up on one end of the chain, it will have no effect.

Parentheses have another function on all daisey cards. When a ground or voltage connection is not requested, parentheses will indicate the source or driving node. The only significance to the user concerning the source is that this is the node used to file or reference the chain on the computer printout. However, when a ground or voltage node is included in the chain, the routine assigns a sequence number. The ground or voltage chains are then listed and referenced under that number. If the parentheses are omitted, the routine assumes the source to be the first node on the card. On a hold card with no G or V connection, if the user desires to designate a source node that is not the first node listed, he must enclose that source node in parentheses. On a hold card with a G or V connection, the first or last node should be enclosed in parentheses to indicate which end the user wishes grounded. Otherwise, the routine will select the end requiring the shortest wire length.

In summary, it is found that parentheses have a dual role. When a ground or voltage connection is desired, parentheses can be used to restrict which end of the chain the ground or voltage connection will be made-provided the node they enclose ends up on one end of the chain. When a ground or voltage connection is not requested, they indicate the node to be taken as the source. Having parentheses around the first node would, of course, be redundant; but acceptable. It also should be mentioned that parentheses are not used as a delimiter, but constitute part of the node field and must still have a space to separate them from other fields.

An important application of parentheses appears when it is necessary to place a capacitive filter on the board. To do so merely requires a daisey chain linking the capacitor and the node requiring the filtered voltage. The voltage request, [VCC] along with parentheses around the node, causes the voltage connection to be wired near the dip rather than near the discrete carrier containing the capacitor.

In addition, the user may name each daisey chain. The name is entered in the daisey chain list as though it were a separate node. To signal the alphanumeric string (in which any character is acceptable save one) the user codes a slash, [/], followed by the name. The string may be up to 12 characters with termination of the field occurring in one of three ways--either the twelfth character, another slash, or the end of the

card. In addition, leading spaces after the first slash are skipped over and not counted. The only character not allowed is, of course, another slash which would prematurely terminate the field.

[END]

This required card entry signals the routine that the last entry on the last data card has been read. The three letters may appear anywhere on any card as a separate input field. However, they have no such effect when located in the name fields of the various cards.

Some of the above input data card options include dip and node subfields. These will now be described for this specific routine called PLUGWRAP.

Dip

Each dip must be referred to by board letter and dip number in that order. Thus, [B21] refers to dip number 21 on board segment "B". If the letter is omitted, the routine will assume an "A", and thus simplify the keypunching when only a single width board is needed. This shortcut is not recommended on multiwidth boards, however, because of the confusion that may result to the keypunch operator.

Node

The nodes have four different forms. They are:

1. I/O pins
2. Dip and pin number nodes
3. Ground or voltage requests
4. Name fields

I/O Pins

Input/output pins have the same appearance as dip fields above. In this case, however, the number following the letter is the I/O pin number. Thus, [C46] refers to I/O pin 46 on board "C". The user may add an asterisk before the letter, if he wishes, to place added significance to an I/O pin, i.e., [*C46]. If the asterisk is used, the routine will assume board "A" when the letter is ommitted. If the asterisk is not used, the routine will signal an error when the letter is omitted. Therefore, there must be at least one nonnumeric character preceding the pin number to signify an I/O pin.

Dip and Pin Number Nodes

Dip and pin number nodes consist of two appropriate numbers separated by a dash and possible preceded by a letter to indicate board segment. Thus, [F21-9] refers to pin 9 of dip 21 on board F. Again, the letter may be ommitted and the routine will assume an "A." This convention is primarily intended for single width boards allowing the keypunch operator to use an asterisk for I/O pins, and two numbers separated by a dash to signify the dip and pin number node.

Ground and Voltage Request

Daisey chains that need to be grounded or tied to VCC will automatically be wired as such if a ground or voltage request is included in the daisey chain entries on the card. The routine accepts any string of letters beginning with a "G" or "V". Most keypunch operators will probably use [G] or [V] as the easiest and quickest form to punch.

It will be recalled that the "GND" and "VCC" card can also be used to accomplish this same purpose. The only difference is that in a daisey chain with a ground or voltage request, all the nodes listed will first be chained together and then one end will be connected to the nearest appropriate "VCC" or "GND." On the "GND" or "VCC" card each node is connected separately to the nearest GND or VCC pin as though each were a separate chain. Finally, the reader is reminded that the half pin number of all dips will be automatically grounded by the PLUGWRAP routine--provided, the pin is not referenced, or the dip is not listed on a "scratch" card.

Name Field

The name field of up to twelve characters is signaled by a slash. The first nonspace character following the slash begins the field which can be terminated in one of three ways--either the twelfth character, another slash, or the end of the card. As an example, the user might punch, [/CLOCK/] on the daisey card. The name would then appear on all output that refers to that chain. There is no restriction on the location of any node subfield, consequently, the name field node can be first, last, in between, or even on a continuation card all by itself. Furthermore, there is only one restriction as to character choice once the field has been started. That restriction is a slash which would prematurely terminate the field.

The author must confess that this restriction was brought to light when he attempted to name a chain "I/O." The choice of field terminator was then reconsidered, but the convenient location of the slash on the keyboard won out over the inconvenience of not being able to use it in a name field.

PLUGWRAP Sample Input Data Deck

A sample input data deck follows. The numbers used are fictitious and do not represent a valid design.

```
NAME PLUGWRAP TEST
NAMEDECK NSWC WA-22
RAYTHEON DECK
DIP16 22 27
/CLOCK/  11-1  17-4  (13-5)  23-13  21-5  28-5  12-8  27-5
     27-4  25-2  16-5
/INPUT/ 25-6  23-10  (*1)  16-2  22-5
11-5  13-1  (23-11)  /OUTPUT
/FILTER/  (23-8)  V  25-1
22-6  27-6  (27-8)  G
25-8  25-9  25-10  25-11  25-12  24-13  25-14  G /DISCRET CAR/
HOLD  (B5-3)  B1-10  B11-6  G
SCRATCH  16  22
VCC  16-10  22-11
GND  11-7  12-7  13-7  16-7  17-7  21-7  22-8  23-12
NO PUNCH
END
```

Chapter 3

GNRLWRAP

This routine handles the general purpose boards, such as Augat type 8136UG1-9 and 813UG1-18. These boards can handle nonstandard dips and/or components. For instance, the user might wish to use a 24 pin, double wide dip. The PLUGWRAP routine discussed in Chapter 2 cannot accommodate this type.

As an alternative to the Augat boards, the routine will also handle the similar type manufactured by Amp Incorporated. The Amp boards are available with the Termi-Point clips which might be an advantage to the user if the board is to be wire wrapped by hand. The Termi-Point tool automatically strips the wire in the process of attaching the clip whereas most hand wire wrapping tools, of reasonable cost, do not. The Amp Termi-Point boards are listed under part number 116193-2 and 116193-6.

To assemble a deck application for the NAVSURFWPNCEN CDC 3200, the user needs the following:

1. Job card [⸗Job,1,XXXX,ND] beginning in column one where XXXX may be the user's identification.
2. Binary Deck called GNRLWRAP
3. Run card [⸗Run] beginning in column one
4. Data deck (described below)
5. End of file card [$\begin{smallmatrix}77\\88\end{smallmatrix}$EOF]
6. Endscope card [⸗ENDSCOPE]

Data Deck

The data deck consists of simple code word instructions along with node to node interconnection data. But first, a few general comments.

1. All input is free field with one or more spaces used as the only delimiter. All eighty columns of the card are read.

2. Since spaces act as a delimiter and terminate a field, none can occur within a field except in the case where a name field has been started.

3. A space in column one signals a continuation of the previous card. There is no limit to the number of continuation cards. The "end of card" also terminates a field and therefore a single node reference may not be split between two cards, i.e., breaks can occur only between nodes.

4. The order of the cards is unimportant with the exception that continuation cards must follow, in any order, behind the lead card they continue.

5. The square brackets, [ ], will be used in this report to indicate what is punched on the card, i.e., everything between the brackets is punched on the card. In some cases, however, the brackets may be eliminated where the meaning is clear (such as the sample input data deck at the end of this chapter).

Data Deck Options

[NAMEDECK XXXXXXXXX]

This card is used by the user who wishes to name a Raytheon drive deck. The nine characters denoted by the X's will be punched in columns 72 to 80 of the Raytheon drive deck. The first nonspace character on or after column 9 begins the 9-character X field.

[NAME XXXXXXXXXXXX?YYY ...,]

This card allows the user to name his circuit design. The first nonspace character on or after column 5 begins the field which extends to the end of the card but is limited to 72 characters maximum. The character string will appear on all output. The Raytheon drive deck, if requested, has room for only the first 12 or 13 characters, however.

[NO PUNCH]

This card is self-explanatory and is used by the user who wishes to check the output results before the actual punching of the Raytheon drive deck. The single space between the "O" and "P" is required.

[RAYTHEON DECK]

This card requests a Raytheon drive deck output format rather than the hand wire listing which is the default mode. The

routine recognizes only the first eight characters and then ignores or skips over the next five. Therefore, just the word [RAYTHEON] would suffice.

[DIPXX Dip1 Dip2 Dip3 ...]

This card is used to list those dips that are other than 14-pin (which is the default mode). The double character XX-field will contain the number of pins of all dips listed on the card. This number may not be larger than 40. Spaces may occur between the P and XX-field, if desired. Following the XX-field, the appropriate dips would be listed (separated from each other by one or more spaces).

In some cases, dips are double wide. These dips must be entered on a dip card with a "D" following the dip number, i.e., like 18D.

[GND Node1 Node2 Node3]

This card makes a separate chain out of every node listed and ties each separately to the closest available ground pin.

[VCC Node1 Node2 Node3]

This card makes a separate chain out of every node listed and ties each separately to the closest available VCC pin.

[ROW Row Row Row]

This card locates the rows of the dip matrix in the event the user wants to force a certain location. The routine automatically locates each row of dips based on all dips in the preceding rows that have 18 pins or less. In addition, the first row is automatically started in pin position 2, if not specified on the card.

The row entry fields consist of two numbers separated by a dash. The first is the row number, counting from the top of the board down; and the second is the pin number of all columns on the board. Pin number one, of each dip in the row, will be located in that column pin number. The only exception to this is when a particular dip is referenced on a "START" card. The "START" card has final say for any one dip, but does not affect the location of the rest of the dips in the row.

[START Start1 Start2 Start3 ...,]

This card is used to force the location of pin number one of any dip. The input start fields, above, consist of a dip number,

dash, board letter, column letter, and column pin number--in that order. An alternate form is also available. This form corresponds to the boards manufactured by Amp Incorporated. Instead of the two letters that follow the dash, the user may use only one where the letters run from "A" to "T" omitting "I" and "O." Thus, letters "A" through "J" indicate the first board, and letters "K" through "T" the second. As an example, [18-M43] and [18-BC43] are equivalent. It is also readily seen that single width boards may always use the single letter, if desired; and it will be correctly interpreted as the proper column letter. (i.e., [18-B43] is equivalent to [18-AB43].)

[AMP BOARD]

This card is used to inform the routine that the Amp board is being used. This causes the proper location to be computed for the I/O pins which are similar but not quite the same as on the Augat board. The Amp board has 50 I/O pins whereas the Augat board has only 46. To be consistent, the first 46 pins on the Amp board are in the same relative location as on the Augat board. That is, looking at the bottom of the board, pins 1 through 23 start from the left on the top I/O pin row. Pins 47 and 48 complete the row on the Amp board and do not exist on the Augat board. The second row starts from the left with 24 through 46 for both boards and ends with 49 and 50 on the Amp board. Furthermore, the interested reader who checks, will note that the Amp board I/O pin rows are closer together and shifted to the left when compared to the Augat board. This information is of little consequence to the user who can usually ignore the differences. However, a double wide Amp board <u>requires</u> the [AMP BOARD] card to force the proper notation on the second board column headings.

[DAISEY Node1 Node2 Node3 ...,]

This card supplies the basic interconnection data to the routine, and since it will probably be used the most, it has been made the default mode of code words. In other words, no code word is also a daisey card as follows.

[Node1 Node2 Node3 ...,]

Here the daisey card is assumed. All nodes listed on the card (and its continuation cards) will be chained together in the most optimum order (minimum total wire length). If this optimum order is not desired, another form of the Daisey card is available (see Hold-card described below).

[HOLD Node1 Node2 Node3 ...,]

On this card the listed nodes will be held "as is" and will not be reordered. This will make the card useful in generating

documentation that agrees with handmade corrections. It might even be desirable to generate a new input drive deck so that reordered boards will be carbon copies of the corrected prototype.

A slight complication exists if a ground or voltage connection is required. On any daisey card (which includes the hold card as an alternate form) ground or voltage connections are made by punching [G] or [V] on the card as a valid node input. In fact, any character string, without numbers, that has G or V as a first character, will be accepted. Thus, the user may prefer [GND] or [VCC]. The problem arises from the fact that the ground or voltage connection may be made from either end of the chain. To specify which end, the user punches enclosing parentheses around the first or last node of the chain, and the ground or voltage connection will be restricted to that end. The user is cautioned that on standard daisey cards (nonhold form) the order of the nodes may be permuted by the routine. Therefore, if the node enclosed by parentheses does not end up on one end of the chain, it will have no effect.

Parentheses have another function on all daisey cards. When a ground or voltage connection is not requested, parentheses will indicate the source or driving node. The only significance to the user concerning the source is that this is the node used to file or reference the chain on the computer printout. However, when a ground or voltage node is included in the chain, the routine assigns a sequence number. The ground or voltage chains are then listed and referenced under that number. If the parentheses are omitted, the routine assumes the source to be the first node on the card. On a hold card with no G or V connection, if the user desires to designate a source node that is not the first node listed, he must enclose that source node in parentheses. On a hold card with a G or V connection, the first or last node should be enclosed in parentheses to indicate which end the user wishes grounded. Otherwise the routine will select the end requiring the shortest wire length.

In summary, it is found that parentheses have a dual role. When a ground or voltage connection is desired, parentheses can be used to restrict which end of the chain the ground or voltage connection will be made--provided the node they enclose ends up on one end of the chain. When a ground or voltage connection is not requested, they indicate the node to be taken as the source. Having parentheses around the first node would, of course, be redundant; but acceptable. It also should be mentioned that parentheses are not used as a delimiter, but constitute part of the node field and must still have a space to separate them from other fields.

An important application of parentheses appears when it is necessary to place a capacitive filter on the board. To do so merely requires a daisey chain linking the capacitor and the node requiring the filtered voltage. The voltage request, [VCC] along with parentheses around the node, causes the voltage connection to be wired near the dip rather than near the discrete carrier containing the capacitor.

In addition, the user may name each daisey chain. The name is entered in the daisey chain list as though it were a separate node. To signal the alphanumeric string (in which any character is acceptable save one), the user codes a slash, [/] , followed by the name. The string may be up to 12 characters with termination of the field occurring in one of three ways--either the twelfth character, another slash, or the end of the card. In addition, leading spaces after the first slash are skipped over and not counted. The only character not allowed is, of course, another slash which would prematurely terminate the field.

[END]

This required card entry signals the routine that the last entry on the last data card has been read. The three letters may appear anywhere on any card as a separate input field. However, they have no such effect when located in the name fields of the various cards.

Some of the input data card options include dip and node subfields. These will now be described for this specific routine called GNRLWRAP.

Dip

First off, the dips themselves are referred to by their array positions, i.e., row and column, as one views the board from the bottom or wire-wrap side. Thus, dip 11 would occupy the upper left position utilizing board pin colums "J" and "H" with pin number 1 of the dip inserted in the lowest pin number of column "H". Dip 21 would be immediately below this position, etc., progressing until no more rows could be placed on the card (a maximum of 10 rows in the case of all 8-pin dips).

The dip columns are numbered from 1 to 9 for a double width board. However, single width boards can accommodate only 4 columns of dips. The fifty pins of board column "A", in this case, are not used; unless, of course, a dip is forced into this column by a "start" card.

Node

The nodes have four different forms. They are:

1. I/O pins
2. Dip and pin number nodes
3. Ground or voltage requests
4. Name fields

I/O Pins

The general layout of the I/O pins has been discussed in conjunction with the "AMP BOARD" input deck card. To refer to them, the pin number will normally be preceded by an asterisk and board letter. However, certain relaxations of form will be accepted.

The asterisk was selected to call attention to the I/O pins and set them off from the dip numbers. However, since two sets of I/O pins exist on a double board, a letter is used to refer to the "B" section. On the other hand, if a letter is used, the routine will not require the asterisk.

In summary, for the I/O pins, the user must have either an asterisk or letter preceding the pin number. Both may be used (in the order, asterisk, letter) but both are not required. If the letter is omitted, board "A" will be assumed.

Dip and Pin Number Nodes

These nodes will be referenced by array number, dash, and dip pin number. The dip numbers can range from 11 to 109, i.e., row one, column one to row ten column nine. The routine can handle dips up to a maximum of 40 pins per dip.

Ground and Voltage Request

Daisey chains that need be grounded or tied to VCC will automatically be wired as such if a ground or voltage request is included in the daisey chain entries on the card. The routine accepts any string of letters beginning with a "G" or "V". Most keypunch operators will probably use G or V as the easiest and quickest form to punch.

It will be recalled that the "GND" and "VCC" card can also be used to accomplish this same purpose. The only difference is that in a daisey chain with a ground or voltage request, all the nodes listed will first be chained together and then one end will be connected to the nearest appropriate "VCC" or "GND." On the "GND" or

"VCC" card each node is connected separately to the nearest GND or VCC pin as though each were a separate chain.

Name Field

The name field of up to twelve characters is signaled by a slash. The first nonspace character following the slash begins the field which can be terminated in one of three ways--either the twelfth character, another slash, or the end of the card. As an example, the user might punch, [/CLOCK/] on the daisey card. The name would then appear on all outputs that refer to that chain. There is no restriction on the location of any node subfield, consequently, the name field node can be first, last, in between, or even on a continuation card all by itself. Furthermore, there is only one restriction as to character choice once the field has been started. That restriction is a slash which would prematurely terminate the field.

The author must confess that this restriction was brought to light when he attempted to name a chain "I/O." The choice of field terminator was then reconsidered, but the convenient location of the slash on the keyboard won out over the inconvenience of not being able to use it in a name field.

GNRLWRAP Sample Input Data Deck

A sample input data deck follows. The numbers used are fictitious and do not represent a valid design.

```
NAME GNRLWRAP TEST
NAMEDECK NSWC WA-22
RAYTHEON DECK
AMP BOARD
DIP24   23D
DIP16   22    27
NO PUNCH
START 23-BC10
ROW  2-14
HOLD   33-5   (35-8)   41-10
/CLOCK/ 11-1   17-4   (13-5)   23-19   21-2   28-5   12-8   27-5
     27-4   26-8   16-5
/INPUT/   25-6   23-20   (*1)   16-2   22-5
11-5   13-1   (23-10)   /OUTPUT
/FILTER/   (23-8)   V   25-1
22-6   27-6   (27-8)   G
25-8   25-9   25-10   25-11   25-12   25-13   25-14   G /DISCRET CAR/
GND   11-7   12-7   13-7   16-7   17-7   21-7   22-8   23-15
25-7   27-7   28-7
VCC   11-14   12-14   13-14   16-14   17-14   21-14   22-16
23-24   26-14   27-14   28-14
END
```

Chapter 4

GENERAL COMMENTS

These routines were written to minimize errors and reduce the tedious, monotonous work of wire list transformations. The routines minimize the human element and, consequently, almost completely eliminate this type of error.

In addition, the routines have been sprinkled with diagnostic checks. These will be self explanatory to the user as a computer abort and diagnostic printout occurs. The user then corrects the offending input data and tries again. However, if the program runs to completion, it is reassuring to know that the following possible errors did not occur.

1. Duplication of single node. It is not legal to reference a single node in more than one chain. To do so would attempt to tie the chains together. This, in turn, could produce smoke if different power supply chains were involved.

2. Wrong format of data input. For instance,

   (a) No spaces can exist within the input node field.

   (b) There is a definite order to each node field such as board letter, dip number, dash, and pin number.

   (c) A left parenthesis requires a right terminating parenthesis.

   (d) Pin numbers must be less than or equal to the total number of pins on the dip (i.e., referencing pin 16 of a 14-pin dip is a no-no).

   (e) The input data card deck cannot request more ground or voltage connections than three times the number of available pins.

   (f) Dips, when located by the GNRLWRAP routine, must lie within board limits. The same holds true for dips located by the user via the "start" card. In addition, dips may not conflict with one another as to position

on the board. Also, there must be a blank board pin between dips in the same column.

(g) Single node chains cannot occur.

In summary, it must be remembered that, by no means has every conceivable error been checked for in the routines. In fact, the routines could surely be improved with the application of more time and money, but somewhere along the way we must stop and use what we have.

Chain Length Minimization

Some readers might be interested in knowing how the chains were ordered so as to minimize their length. Chain length minimization is good from the standpoint of less wire buildup on the board and its lower associated cross coupling capacitance.

To begin with, the author must confess that he saw one of those TV ads where the soap opera-ee exclaimed, "Mother, let me do it myself." Anyway, the idea caught hold and the algorithm was developed without turning to the literature. Most certainly the solution lies buried somewhere in a textbook.

To minimize a chain of nodes, without checking all N-factorial permutations, the algorithm starts small. That is, assuming the chain to have more than three nodes, it simply takes three nodes at random and puts them in the best order (there are only three ways to check). Then the remaining nodes of the chain are inserted in the best place, one at a time. In this way, the chain grows like a crystal until it encompasses all of the nodes. This simple approach was found to work quite well although it could be improved with minor modification.

First, after each additional node is added, the partial chain is checked for the possible improvement that would result if each node were to be reinserted on the opposite side of every other node. If an improvement exists, the insertion is made, and a new partial chain becomes the basis to which the next node is added. When the chain has grown to the total number of nodes, the length and order is stored. The whole process is then repeated starting with three, presumably different, nodes. If the new order has less length than the previous order, the new order replaces the saved, etc. The reiteration process continues until there has been a sufficient number of tries without improvement. The sufficient number of tries, by the way, has been more or less arbitrarily set to the number of nodes in the chain. It could just as well, perhaps, be set to a constant--say five.

The routine is quite fast when compared to the time of making N-factorial computations of chain length (especially when "N" is large, say 50). In the case of small "N", it may be slightly slower

than the N-factorial way; but then, either way is short and it does not make much difference to the user waiting for his results.

Interfacing With Other Routines

Some users may wish to incorporate the features of PLUGWRAP and GNRLWRAP in their own routines. Therefore, a brief discussion of the routine architecture, input requirements, and machine dependence is in order.

To begin with, the routines are heavily committed to character mode variables. These variables consist of a single character (6 bits) and are stored four to a computer word (24 bits). As a consequence, considerable modification would be required to use the routines on a machine (other than the CDC 3200) that did not have character mode capability. Furthermore, there is considerable variable length I/O formating geared to the 24-bit computer word.

The routines are written in Fortran and, therefore, could be modified for another machine in a reasonable length of time. Another complication that may cause a problem to the user is the fact that there is some character decoding based on the internal BCD code of the character. For instance, the internal BCD code (on the CDC 3200) of all numbers is less than or equal to nine. This feature was used in one of the routines to skip over all nonnumerical characters. In another application, the characters were directly decoded through knowledge of their internal BCD code. For instance, the internal BCD code of a single digit number is equal to the number; and the internal BCD code of the first six letters of the alphabet is equal to the letter rank or position plus sixteen. Thus, the third letter of the alphabet has an internal BCD code of $19 = 23_8$. This feature was quite useful in converting board and column letters to horizontal coordinates.

Both routines are structured around two arrays--one, the input data array, and two, the board coordinate array. The first array contains data as read or determined from the input data cards. The second array contains x and y coordinates. In addition, the data contained in the first array is used as the subscripts (or addresses) of the data in the second.

Each chain is stored one on top of the other in the first array with another small array keeping track of the beginning and end. Later on, the ordering algorithm is called successively for each chain. This, in turn, flips the addresses around until the desired wiring order has been achieved.

From this brief description, the interested reader can probably determine whether or not his own software is compatible to form a direct interface link. In most cases here at NAVSURFWPNCEN, it will

probably be best to generate an input data card deck and use the CDC 3200.

Additional information is available from the author on an informal basis. In addition, source listings may be obtained through Navy channels.

DISTRIBUTION

	Copies
Defense Documentation Center Cameron Station Alexandria, VA 22314	12